Abstract for the Testcase C2.3c
by Ralf Hartmann, DLR, Braunschweig

1 Code description

The PADGE code implements higher order discontinuous Galerkin methods on unstructured possibly curved and locally adapted meshes.
Parameter settings for the current test case:

- Discretization/Higher order capability:
  - Legendre polynomial basis functions of polynomial degrees 0-3.
  - Local Lax-Friedrichs flux
  - BR2 scheme
  - Characteristic farfield boundary conditions
- Solver/Parallel capability
  - Implicit solver (Backward Euler)
  - Residual and matrix assembly is parallelized
  - Linear solver: GMRES with ILU preconditioner (parallelized by PETSc)

2 Case summary

- The density residual has been reduced to 10^{-10} relative to the freestream density residual.
- Computations have been performed on old nodes of the $C^2A^2S^2E$ cluster (86 compute nodes with 8 cores (AMD Opteron, 1.9GHz QuadCore processors)). The number of cores used ranged between 1 and 12. TAUBench was between 19.069 and 19.768 sec. p=1 on the finest mesh has been computed (with different solver setting) on the ANTON cluster (68 compute nodes with 8 cores (two AMD Opteron, 2.7 GHz, quad core processors, Typ 2384, Shanghai)) where 16 cores were used with an TAUBench of 12.903 sec.

3 Meshes

The q4 meshes as provided on the HOW webpage

- btc0-NLR-T2.v2.m4.msh 6,656 cells
- btc0-NLR-T3.v2.m4.msh 53,248 cells
- btc0-NLR-T4.v2.m4.msh 425,984 cells

Note, that the btc0-NLR-T1.v2.m4.msh mesh with 768 cells is obmitted in the results. For p=1 the solver struggled but finally converged, for p=2 the solver did not converge at all. After all, a mesh with 768 cells is far too coarse for a turbulent flow.

4 Results

Reference values (taken from [HHL11][1]): $C_d^{\text{ref}} = 0.00835$, and $C_l^{\text{ref}} = 0.006612$.

[1] R. Hartmann, J. Held and T. Leicht Adjoint-based error estimation and adaptive mesh refinement for the RANS and k-ω turbulence model equations J. Comput. Phys., 230(11): 4268-4284, 2011.

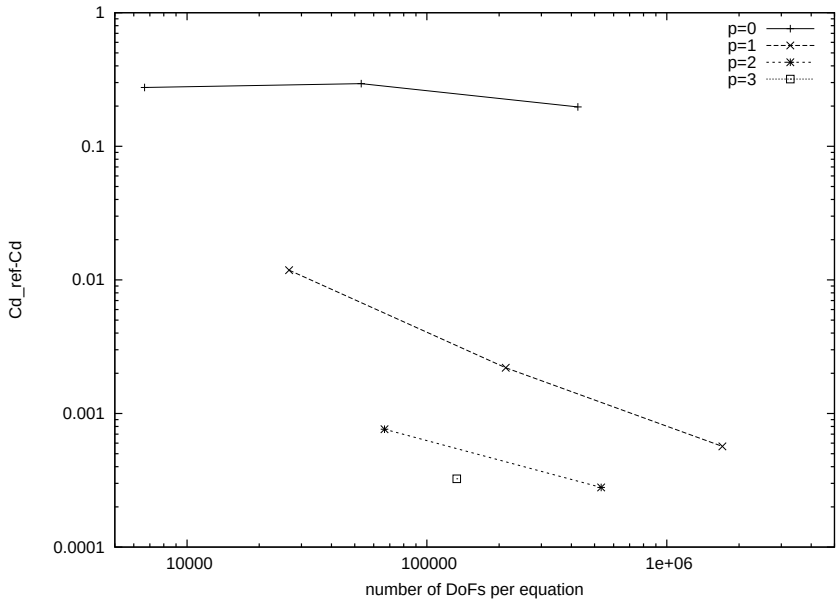

Figure 1: Error in C_d vs. number of DoFs per equation

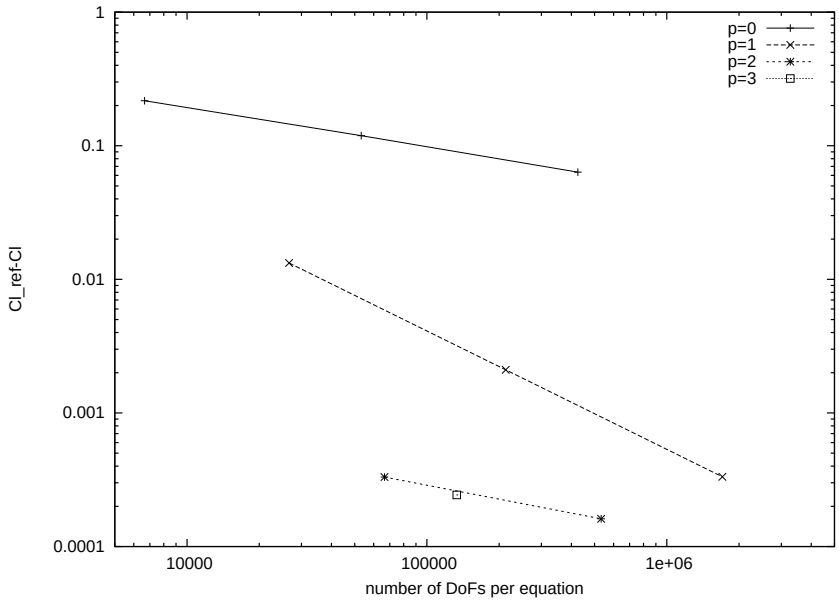

Figure 2: Error in C_l vs. number of DoFs per equation

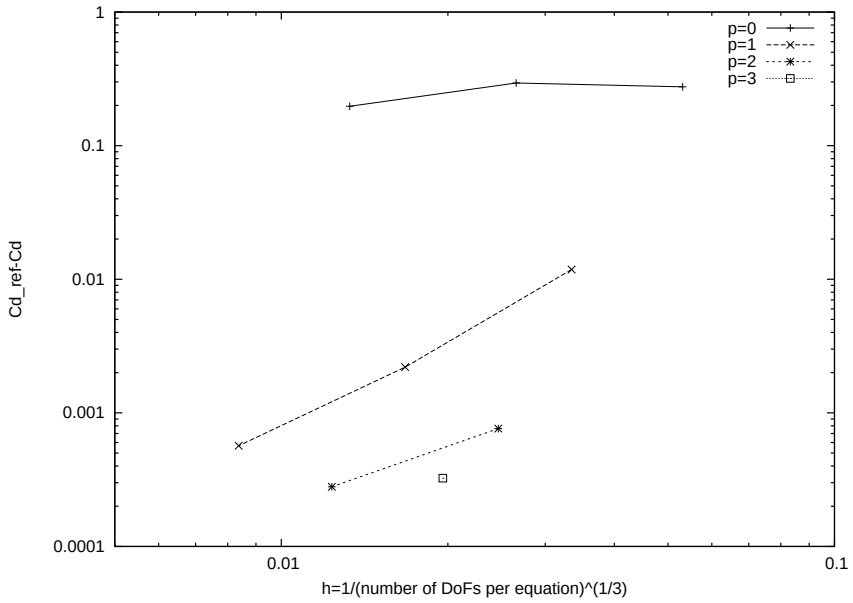

Figure 3: Error in C_d vs. $h = 1/\sqrt[3]{\text{number of DoFs per equation}}$

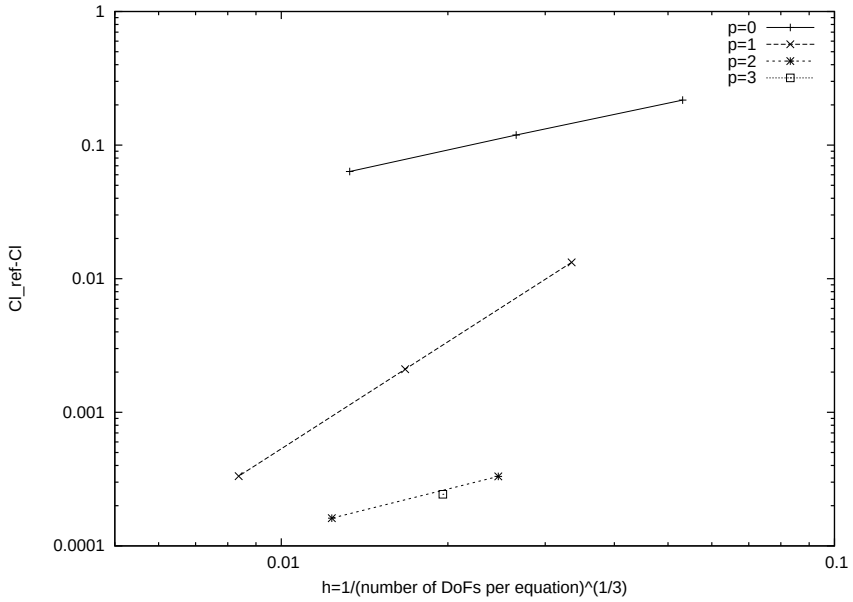

Figure 4: Error in C_l vs. $h = 1/\sqrt[3]{\text{number of DoFs per equation}}$